

RETRIEVING PERFORMANCE DATA FROM DEVICES IN A STORAGE AREA NETWORK

1 **Technical Field**

2 The technical field relates to storage area networks, and, in particular, to
3 improvements in monitoring the performance of a storage area network.

4 **Background**

5 A storage area network (SAN) is a network of elements including multiple
6 servers, multiple storage devices (such as disk arrays and tape libraries), and a high-speed
7 network of interconnecting elements (such as fiber channels, switches, hubs, and bridges)
8 that establishes direct and indirect connections between the servers, between the storage
9 devices, and between the servers and the storage devices.

10 SANs are, however, typically made of heterogeneous modules. As a result of
11 their versatility and popularity, a number of different SAN devices exist on the market,
12 each having its own protocols and requirements. Further, current SAN management tools
13 are often vendor-dependent and system-dependent. Since each SAN device may be
14 produced by a different vendor, and each SAN device may have its own proprietary
15 communication systems and information, the management of SAN resources is a difficult
16 task for storage administrators.

17 **Summary**

18 A method for using a performance interface to retrieve performance data from
19 SAN devices in a storage area network (SAN) includes instructing a device plug-in (DPI)
20 to retrieve performance metrics data from a corresponding SAN device, determining a
21 minimum polling interval for polling the SAN device for the performance metrics data,
22 determining a maximum polling interval for polling the SAN device for the performance
23 metrics data, and collecting the performance metrics data from the DPI using the
24 performance interface.

25 **Description of the Drawings**

26 The preferred embodiments of the method and apparatus for retrieving
27 performance data from SAN devices in a local area network (SAN) will be described in
28 detail with reference to the following figures, in which like numerals refer to like
29 elements, and wherein:

30 Figure 1 illustrates an exemplary system for retrieving performance data from
31 SAN devices in a SAN;

Figure 2 illustrates an exemplary device plug-in (DPI) that is used in the exemplary system of Figure 1 for retrieving performance data;

Figure 3 is a flowchart illustrating an exemplary method for using a performance interface to retrieve performance data from SAN devices in a SAN; and

Figure 4 illustrates exemplary hardware components of a computer that may be used in connection with the exemplary method for using a performance interface to retrieve performance data from SAN devices in a SAN.

Detailed Description

A system and method for retrieving performance data from different devices in a storage area network (SAN) use a performance interface to communicate effectively with different types of SAN devices in a heterogeneous SAN environment. The performance interface instructs a device plug-in (DPI) associated with each SAN device to retrieve performance metrics data from the corresponding SAN device. The performance interface determines a minimum polling interval and a maximum polling interval for polling each SAN device for the performance metrics data. The performance interface further provides each DPI with an address of the corresponding SAN device.

The performance interface may be implemented with a storage area manager (SAM) and a storage optimizer, such as the OPENVIEW[®] SAM and storage optimizer. The SAM supports enterprise storage utility services with integrated tools that help information technology (IT) departments reduce storage management costs, protect existing storage investments, efficiently utilize resources, and deliver guaranteed quality of service to customers, partners, and employees. The SAM enables system administrators to simplify and automate management of multivendor storage resources across disk, tape, direct-attach, and networked storage infrastructures. The SAM can also centrally manage and monitor availability, performance, usage, growth, and cost across a distributed enterprise. Further, the SAM enables system administrators to optimize resource utilization and operations, and to seamlessly integrate storage and storage services with the enterprise-wide IT service management system.

The storage optimizer may, from a single management station, monitor the performance of storage network components, including hosts, infrastructure and storage. The storage optimizer also collects data and provides a complete reporting structure for evaluating, monitoring and managing the quality and cost of IT services. System administrators can receive automatic notification of impending performance problems

1 before the problems become serious, and can track performance of the system
2 components over time. The storage optimizer helps system administrators predict
3 problems and improve efficiencies while optimizing investment in networked storage.
4 The combination of performance monitoring, historical tracking, and trend analysis
5 makes the storage optimizer a powerful tool for analyzing and upgrading storage
6 infrastructure.

7 Figure 1 illustrates an exemplary system for retrieving performance data from
8 SAN devices in a SAN. Referring to Figure 1, a SAN 110 includes several SAN devices
9 120, such as storage arrays, and corresponding device plug-ins (DPIs) 130 for each SAN
10 device 120. A DPI 130 is Java code specifically written to the specifications of the SAN
11 management software. The DPI 130 allows the corresponding SAN device 120 to
12 communicate with a SAN management application 160. The DPI 130 is specific to each
13 SAN device 120 and may be developed by any software developer. The DPI 130 may
14 communicate with the corresponding SAN device 120 directly using, for example, Simple
15 Network Management Protocol (SNMP). Other SAN devices 120 may require a separate
16 host machine 125 to communicate with the corresponding DPIs 130. The host machine
17 125 typically maintains a database (not shown) that stores performance data for the SAN
18 device 120.

19 With continued reference to Figure 1, the SAN 110 is managed by the SAN
20 management application 160. The SAN management application 160 includes
21 corresponding abstract data sources 150 for each SAN device 120 and a performance
22 application 170. The abstract data source 150 may be Java code. The performance
23 application 170 collects performance metrics data from the SAN devices 120 in the SAN
24 110. Examples of performance metrics data include a number of device write operations,
25 read operations, and cache hits. Each abstract data source 150 within the SAN
26 management application 160 has a one-to-one relationship with each SAN device 120.
27 The abstract data source 150 uses the methods defined within performance interface 140
28 (shown in Figure 2) to communicate with the SAN device 120 through the corresponding
29 DPI 130.

30 Figure 2 illustrates an exemplary DPI 130 that is used in the exemplary system of
31 Figure 1 for retrieving performance data. The DPI 130 is a building block of interfaces,
32 such as the performance interface 140, an interface 141 for reporting port information,
33 and an interface 142 for reporting internal disk information. The performance interface
34 140 is an interface between the DPI 130 and the corresponding abstract data sources 150.

1 As noted above, each abstract data source 150 has a one-to-one relationship with each
2 SAN device 120. When the abstract data source 150 makes a request for performance
3 data, the DPI 130 implements methods (i.e., Java code) in the performance interface 140
4 to retrieve the performance metrics data from the corresponding SAN device 120.
5 Different SAN devices 120 provide performance data in different formats. The
6 performance interface 140 provides a single method the performance application 170 can
7 call to retrieve performance metrics data from the SAN devices 120 in a SAN 110.

8 The performance interface 140 includes a function indicator 145 that instructs the
9 DPI 130 to retrieve performance data from the corresponding SAN device 120. When the
10 performance application 170 collects performance metric data from a given SAN device
11 120, the corresponding abstract data source 150 may make a request to the corresponding
12 DPI 130. The DPI 130 implements the methods in the performance interface 140, which
13 instructs the DPI 130 to retrieve the requisite performance metric data from the
14 corresponding SAN device 120. The performance interface 140 allows the individual DPI
15 130 to utilize whatever means necessary to gather the information from the corresponding
16 SAN device 120. For example, the DPI 130 may perform a task to collect performance
17 data. Examples of the task include: reading log files maintained by a SAN device 120,
18 navigating the structure of internal counters maintained by the SAN device 120, and
19 implementing specific application programming interface (API) calls into management
20 software for the SAN device 120.

21 The performance interface 140 further includes a minimum polling indicator 146
22 and a maximum polling indicator 147. The performance application 170 may
23 periodically “poll” all SAN devices 120 to obtain performance metric data. The
24 minimum polling indicator 146 determines a minimum polling interval for polling a SAN
25 device 120 for performance data. The minimum polling interval is the shortest time
26 between performance metric collection that should occur. The minimum polling interval
27 may vary from SAN device 120 to SAN device 120 according to the rate a SAN device
28 120 updates its performance data. The minimum polling interval may be a matter of
29 seconds (such as for an Interconnect device) or as long as an hour (such as for a disk array
30 device).

31 The maximum polling indicator 147 determines a maximum polling interval for
32 polling a SAN device 120 for performance data. The maximum polling interval is the
33 longest time between performance metric collection that should be allowed to occur. The
34 maximum polling interval may vary from SAN device 120 to SAN device 120 similar to

1 the minimum polling interval. The performance interface 140 determines the minimum
2 polling interval and the maximum polling interval for each SAN device 120, enabling
3 each SAN device 120 to be accurately polled by the performance application 170 at a rate
4 between the minimum and the maximum polling intervals.

5 The performance interface 140 further includes an address indicator 148 that
6 automatically provides the DPI 130 with the address of the SAN device 120. For
7 example, if the SAN device 120 communicates with the corresponding DPI 130 through a
8 host machine, the host machine 125 typically maintains the database that stores
9 performance data for the corresponding SAN device 120. The performance interface 140
10 may provide the DPI 130 with the address on the host machine 125 to retrieve the
11 performance metrics data for the corresponding SAN device 120. The performance
12 application 170 does not need to “know” whether a DPI 130 communicates with the
13 corresponding SAN device 120 directly or through a host machine 125.

14 Figure 3 is a flowchart illustrating an exemplary method for using a performance
15 interface to retrieve performance data from SAN devices 120 in a SAN 110. The
16 performance interface 140 determines a minimum polling interval and a maximum
17 polling interval for polling each SAN device 120 for the performance metrics data (block
18 310). After establishing the polling interval (block 320), the performance interface 140
19 provides each DPI 130 with the address of the corresponding SAN device 120 (block
20 330). Next, the performance interface 140 instructs the DPI 130 to retrieve performance
21 metrics data from the corresponding SAN device 120 (block 350). The DPI 130 may
22 perform a task to collect the performance metrics data. The task may include reading log
23 files maintained by a SAN device 120, navigating the structure of internal counters
24 maintained by the SAN device 120, and implementing specific application programming
25 interface (API) calls into management software for the SAN device 120. The
26 performance application 170 then collects the performance metrics data using the
27 performance interface 140 (block 360).

28 Figure 4 illustrates exemplary hardware components of a computer 400 that may
29 be used in connection with the method for retrieving performance data from SAN devices
30 120 in a SAN 110. The computer 400 includes a connection 420 with a network 418 such
31 as the Internet or other type of computer or telephone network. The computer 400
32 typically includes a memory 402, a secondary storage device 412, a processor 414, an
33 input device 416, a display device 410, and an output device 408.

1 The memory 402 may include random access memory (RAM) or similar types of
2 memory. The secondary storage device 412 may include a hard disk drive, floppy disk
3 drive, CD-ROM drive, or other types of non-volatile data storage, and may correspond
4 with various databases or other resources. The processor 414 may execute information
5 stored in the memory 402, the secondary storage 412, or received from the Internet or
6 other network 418. The input device 416 may include any device for entering data into
7 the computer 400, such as a keyboard, keypad, cursor-control device, touch-screen
8 (possibly with a stylus), or microphone. The display device 410 may include any type of
9 device for presenting visual image, such as, for example, a computer monitor, flat-screen
10 display, or display panel. The output device 408 may include any type of device for
11 presenting data in hard copy format, such as a printer, and other types of output devices
12 including speakers or any device for providing data in audio form. The computer 400 can
13 possibly include multiple input devices, output devices, and display devices.

14 Although the computer 400 is depicted with various components, one skilled in
15 the art will appreciate that the computer 400 can contain additional or different
16 components. In addition, although aspects of an implementation consistent with the
17 method for retrieving performance data from SAN devices 120 in a SAN 110 are
18 described as being stored in memory, one skilled in the art will appreciate that these
19 aspects can also be stored on or read from other types of computer program products or
20 computer-readable media, such as secondary storage devices, including hard disks, floppy
21 disks, or CD-ROM; a carrier wave from the Internet or other network; or other forms of
22 RAM or ROM. The computer-readable media may include instructions for controlling
23 the computer 400 to perform a particular method.

24 While the system and method for retrieving performance data from SAN devices
25 in a SAN have been described in connection with an exemplary embodiment, those
26 skilled in the art will understand that many modifications in light of these teachings are
27 possible, and this application is intended to cover variations thereof.